// Dbconnectivity.cpp code to connct

#include<iostream>

#include <stdio.h>

#include <sqlite3.h>

int main(int argc, char\* argv[])

{

sqlite3 \*db;

//Declare variables

char \*MsgErr = 0;

int cr;

cr = sqlite3\_open("database.db", &db);

if (cr)

{

fprintf(stderr, "Cannot open database: %s\n", sqlite3\_errmsg(db));

//Return 0

return(0);

}

//If it is present

else

{

fprintf(stderr, "Database successfully Connected\n");

}

//Close

sqlite3\_close(db);

}

//c code to CreateTable.cpp

#include <stdio.h>

#include <stdlib.h>

#include <sqlite3.h>

static int Call(void \*NotUsed, int argc, char \*\*argv, char \*\*azColName)

{

int i;

for (i = 0; i<argc; i++)

{

//Display message

printf("%s = %s\n", azColName[i], argv[i] ? argv[i] : "NULL");

}

printf("\n");

//Retiurn 0

return 0;

}

int main(int argc, char\* argv[])

{

//Define db

sqlite3 \*db;

char \*MsgErr = 0;

int cr;

char \*sql;

// Open database

cr = sqlite3\_open("test.db", &db);

if (cr)

{

fprintf(stderr, "Cannot open database: %s\n", sqlite3\_errmsg(db));

//Return 0

return(0);

}

//If database present

else

{

fprintf(stdout, "Successfull!!\n");

}

// Create table Veterinarians

sql = "CREATE TABLE Veterinarians (" \

"vid INT PRIMARY KEY NOT NULL," \

"elevel INT NOT NULL);";

sql = "CREATE TABLE Examine(" \

"vid INT PRIMARY KEY NOT NULL," \

"did INT NOT NULL," \

"FEE INT );";

sql = "CREATE TABLE Dogs(" \

"did INT PRIMARY KEY NOT NULL," \

"age INT NOT NULL);";

cr = sqlite3\_exec(db, sql, Call, 0, &MsgErr);

if (cr != SQLITE\_OK)

{

//Display message

fprintf(stderr, "SQL error: %s\n", MsgErr);

//Call sqlite3\_free()

sqlite3\_free(MsgErr);

}

else

{

fprintf(stdout, " successfull!!!\n");

}

//Call sqlite3\_close

sqlite3\_close(db);

//Return 0

return 0;

}

//this code will Insert data into tables

#include <stdio.h>

#include <stdlib.h>

#include <sqlite3.h>

static int CALL(void \*NotUsed, int argc, char \*\*argv, char \*\*azColName)

{

int i;

for (i = 0; i<argc; i++)

{

//Display message

printf("%s = %s\n", azColName[i], argv[i] ? argv[i] : "NULL");

}

printf("\n");

return 0;

}

int main(int argc, char\* argv[])

{

//Defining th database

sqlite3 \*db;

char \*MsgErr = 0;

int cr;

char \*sql;

// Open database

cr = sqlite3\_open("test.db", &db);

if (cr)

{

fprintf(stderr, "Cannot open database: %s\n", sqlite3\_errmsg(db));

return(0);

}

//this willm check If database is present

else

{

fprintf(stderr, "Successfull!!1\n");

}

// Defne Insert statements

sql = "INSERT INTO Veterinarians (vid,elevel)VALUES (1001,1); " \

"INSERT INTO Examine (vid,did,FEE)VALUES (1001,1234,25 ); " \

"INSERT INTO Dogs (did,age)VALUES (1234,10);"

// Execute SQL statement

cr = sqlite3\_exec(db, sql, callback, 0, &MsgErr);

if (cr != SQLITE\_OK)

{

//Display message

fprintf(stderr, "SQL error: %s\n", MsgErr);

sqlite3\_free(MsgErr);

}

else

{

fprintf(stdout, "Successfull\n");

}

sqlite3\_close(db);

//return 0

return 0;

}

// th codes blow will Display data within th databas

#include <stdio.h>

#include <stdlib.h>

#include <sqlite3.h>

//Define Call()

static int Call(void \*data, int argc, char \*\*argv, char \*\*azColName)

{

int i;

fprintf(stderr, "%s: ", (const char\*)data);

//Loop until argc value

for (i = 0; i<argc; i++)

{

printf("%s = %s\n", azColName[i], argv[i] ? argv[i] : "NULL");

}

//Display newline

printf("\n");

//Return 0

return 0;

}

int main(int argc, char\* argv[])

{

sqlite3 \*db;

//Declare variables

char \*MsgErr = 0;

int cr;

char \*sql;

const char\* data = "Call function is called";

// Open database

cr = sqlite3\_open("test.db", &db);

if (cr)

{

//Display message

fprintf(stderr, "Cannot open database: %s\n", sqlite3\_errmsg(db));

//Return 0

return(0);

}

else

{

//Display message

fprintf(stderr, "Successfull!!!\n");

}

// Display data

sql = SELECT Dogs.age, COUNT(\*), AVG(X.fee),AVG(V.eLevel) FROM Dogs AS D,Exmaine AS X, Veterinarians AS VWHERE D.did = X.did AND X.vid = V.vid GROUP BY D.age;

// Execute SQL statement

cr = sqlite3\_exec(db, sql, Call, (void\*)data, &MsgErr);

if (cr != SQLITE\_OK)

{

//Display message

fprintf(stderr, "SQL error: %s\n", MsgErr);

sqlite3\_free(MsgErr);

}

else

{

//Display message

fprintf(stdout, "Successfull\n");

}

//Close th database

sqlite3\_close(db);

return 0;}